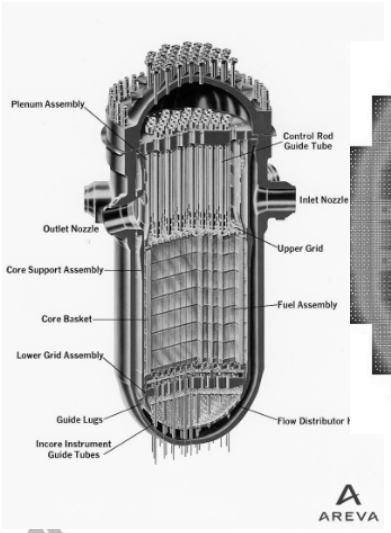
Argonne NATIONAL LABORATORY

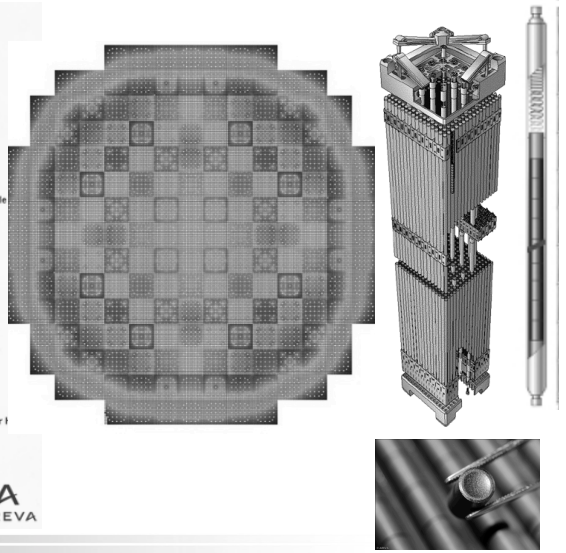
Center for Exascale Simulation of Advanced Reactors: co-design challenges for Monte Carlo neutron transport

Andrew Siegel, ANL
Kord Smith, MIT
Paul Romano, MIT
Ben Forget, MIT

U.S. DEPARTMENT OF ENERGY

LWR Challenge: Predict Pellet-by-Pellet Power Densities and Nuclide Inventories for the Full Life of Reactor Fuel (~5 years)





Three key issues are focus of this tutorial

1. Efficient decomposition algorithms for tallies
2. Efficient decomposition algorithms for cross-section lookup tables
3. Efficient algorithms for on-node parallelism

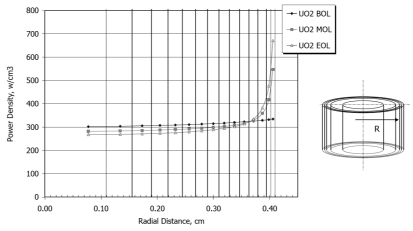
- Each is explained in depth in subsequent slides!

At high level MC algorithm very simple

```
Initialize initial neutron positions
for each batch
  for each particle in batch
    while (not absorbed)
      move particle to next interaction point
      lookup material at collision point
      for each nuclide in material
        for each reaction type
          look up micro cross-section
          build macro cross section
      sample reaction     // either collision or absorption
    end
    sample if fission occurred //guaranteed absorbed here
    if fission
      - tally //one type of tally, others possible
      - add new source sites
  end
  resample source sites //for steady state calculation
  estimate eigenvalue
end
```

The Scale of Monte Carlo LWR Problem – tally memory

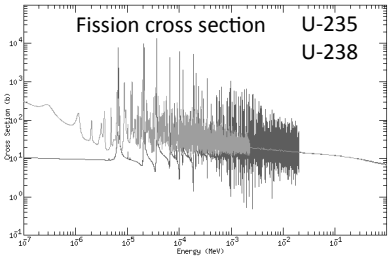
- Detailed spatial tallies required to calculate fuel isotopic inventories
- For a robust reactor simulation, tally data for one fixed point calculation is ~1Tb
- Efficient decomposition methods are needed at exascale



Estimate of size	Property
~200	Fuel assemblies
~700,000	Discrete fuel pins
~35,000,000	Discrete fuel pellets
~350,000,000	Discrete depletion zones
~1,000,000,000,000	Bytes of tally data for 300 nuclides
~100,000,000,000,000	Bytes of tally data for fuel history

The Scale of Monte Carlo LWR Problem – cross-section memory

- Particle tracking requires cross-section lookup at each interaction or change of material region
- Cross-section value depends on energy, nuclide, reaction type, and temperature
- This results in very large lookup tables that need to be read per particle per interaction (tenths of milliseconds)



Estimate of size	Property
~100,000	Cross section energy levels
300-400	Nuclides in fuel region
~50-100	Discrete temperature values
5-10	Reaction types
~300,000,000,000	Bytes of cross section data

The Scale of Monte Carlo LWR Problem – tracking rate

- Target accuracy for reactor analysis requires billions of particles
- Thus, reducing time to solution at exascale is a critical focus area
- This goes hand and hand with data decomposition choices
  - Potentially longer tracking times
- Scalable algorithms/hardware for on-node parallelism critical to success of Monte Carlo at exascale

Estimate of size	Quantity
<= 1.0%	Statistical uncertainty (2-sigma) of tallies
~ 20	Outer iterations (batches)
~ 300	Tracking rate (particles/sec) with current algorithms
~ 25,000,000,000	Particles simulated per batch
~ 300,000,000,000	Bytes of cross section data to access
~ 500,000	Core-hours to calculate one state point

Comment on classic parallelism

- Independent trajectories makes algorithm nearly embarrassingly parallel
  - Romano et. al demonstrated scaling to 100K nodes
  - Synchronization of tally/source needs careful treatment
  - Also, load balancing considerations, small effect in general
- Scalability however assumes replication of tally/x-section data across all nodes!
  - This is in general not possible for full target simulation!!

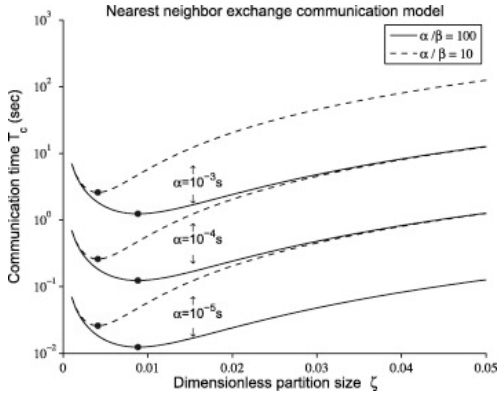
Three major issues at exascale

- Efficient decomposition strategy for 1Tb tally data
- Efficient decomposition strategy for 300Gb cross section data
- Efficient on-node threading by particle history
  - or, for SIMD, algorithm to expose SIMD parallelism

Co-design for large tallies

- Spatial domain decomposition
  - Each spatial region stores tallies within its domain
  - Dramatically reduced memory footprint
  - Must move particles at processor boundaries
    - Nearest neighbor exchanges only
    - However, very high leakage rates
    - What are required network characteristics for efficient data exchange (next slide)
    - What is the impact of load imbalances on performance and cost of resolution on exascale-type machine (subsequent slide)
- Arbitrary decomposition
  - Write tally data to remote processor
  - "tally server" model one implementation
  - What are required interconnect characteristics? (subsequent slide)
  - Hardware support for fast non-blocking write operations?

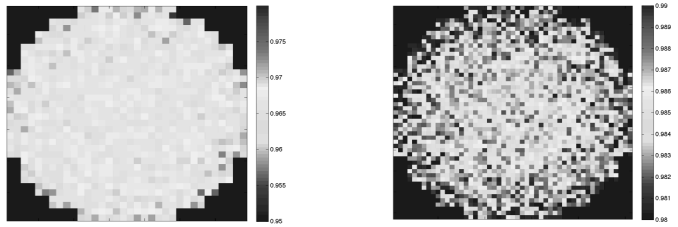
Analysis of Inter-node communication requirements for domain decomposed model: ***MCCK kernel***



A. Siegel, K. Smith, P. Fischer, and V. Mahadevan. **Analysis of communication costs for domain decomposed Monte Carlo methods in nuclear reactor analysis**. *Journal of Computational Physics*, 231(8):3119–3125, April 2012.

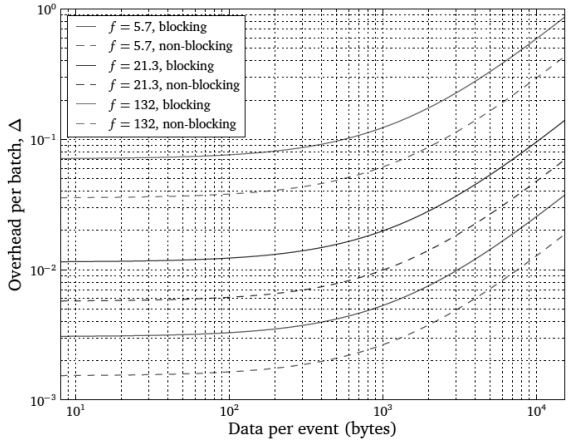
Analysis of impact of load imbalances on domain decomposed model

- Modest effect for large sub-domains
- Predict penalty up to 30x for 1/9th assembly domain size



A. Siegel, K. Smith, P. K. Romano, B. Forget, and K. Felker. **The effect of load imbalances on the performance of Monte Carlo codes in LWR analysis**. *Journal of Computational Physics*, 2012, DOI 10.1016/j.jcp.2012.06.012

Measurements of overhead for tally server model



Paul K. Romano, A. Siegel, B. Forget, K. Smith. **Data Decomposition of Monte Carlo Particle transport simulations via tally servers**. *Journal of Computational Physics*, 2012 (submitted)

Three major issues at exascale

- Efficient decomposition strategy for 1Tb tally data
- Efficient decomposition strategy for 300Gb cross section data
- Efficient on-node threading by particle history
  - or, for SIMD, algorithm to expose SIMD parallelism

Efficient decomposition strategies for cross sections

- Cross section data size:
  - ~2 G-byte for 300 isotopes at one temperature
  - ~200 G-byte for tabulation over 300K-2500K in 25K intervals
    - Data is static during all calculations
    - Exceeds node memory of anticipated machines?
      - Especially when competing with other data structures
- NV-Ram Potential?
  - Data is static during all simulations
    - Size NV-RAM needed depends on data tabulation or expansion approach
    - Static data beckons for non-volatile storage to reduce power requirements
    - Access rate needs to be very high for efficient particle tracking

Efficient decomposition strategies for cross sections

- Fully replace lookup with FLOP/s
  - Cullen's method to compute cross section integral directly from 0^0K data, or
  - Stochastically sample thermal motion physics to compute broadened data
    - Never store temperature-dependent data, only the 0^0K data
    - Cache misses will be much smaller than with tabularized data
    - Flop requirement may be large, but it is easily vectorizable
- Data compression
  - U of Michigan has shown that 20-term expansion may be acceptable
  - ~40 G-byte for 300 isotopes
    - Large manpower effort to preprocess data
    - Many cache misses because data is randomly accessed during simulations

Efficient decomposition strategies for cross sections

- Energy domain decomposition? ***EBMS** kernel*
  - Split energy range into a small number (~5-20) energy groups
  - Bank group-to-group scattering sites when neutrons leave a domain
  - Exhaust particle bank for one domain before moving to next domain
  - Use server nodes to move cross section only for the active domain
    - Modest effort to restructure simulation codes
    - Cache misses will be much smaller than with full range tabularized data
    - Communication requirements can be reduced by employing large particle batches

Three major issues at exascale

- Efficient decomposition strategy for 1Tb tally data
- Efficient decomposition strategy for 300Gb cross section data
- Efficient on-node threading by particle history
  - or, for SIMD, algorithm to expose SIMD parallelism

Efficient on-node parallelism

- Both coarse and fine-grained threading possible
  - Coarse: thread particle loop
  - Fine: thread nuclide macro cross section loop
  - Hybrid: both in same situation (dynamic?)
- Algorithm is inherently scalable but
  - Current multicore machines with current programming models show scaling degradation (see paper and next slide)

A. Siegel, P. Romano, K. Smith, B. Forget, K, Felker. **Multicore performance studies of neutral particle Monte Carlo methods**. *International Journal of High Performance Computing Applications*, 2012 (in preparation)